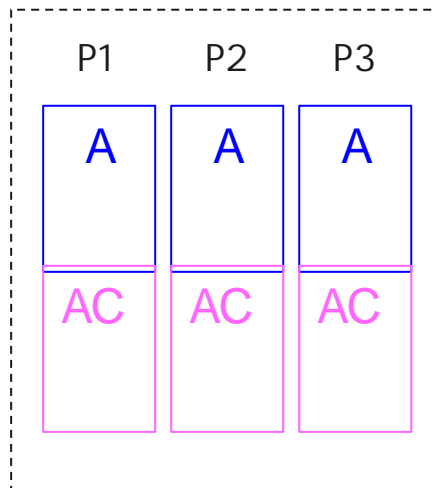


Discussion on possible coupled architecture

1. One executable with sequential execution of components



Sequential execution:

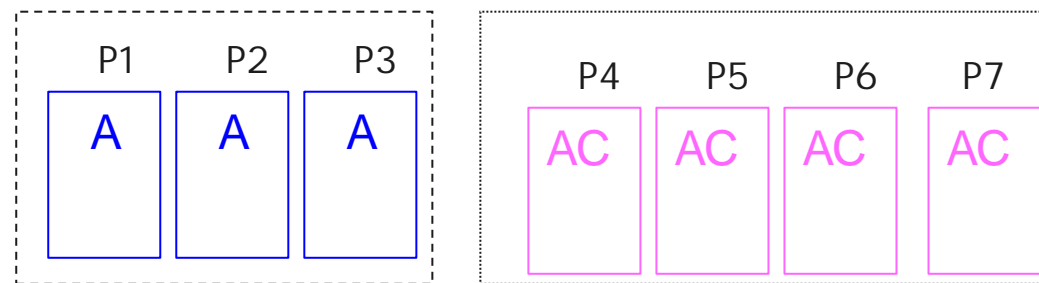
- > optimal for load balancing and throughput
if components are sequential by construction
- > well adapted
if components have same grid and decomposition
- > needs decomposition in sequential sub-components
if components not naturally sequential

One executable:

- > easier to manage for the OS (?)
- > possible I/O conflicts
- > possible communicator conflicts if not given by coupler
- > memory sharing (MS) possible
not flexible but efficient
- > memory copy (MC) possible
flexible, more efficient on most platforms than MP, but less efficient
and requires more memory than MS
- > message passing (MP) possible
flexible, but less efficient on most platforms than MC

Discussion on possible coupled architecture

2. Different executables with parallel execution



Parallel execution:

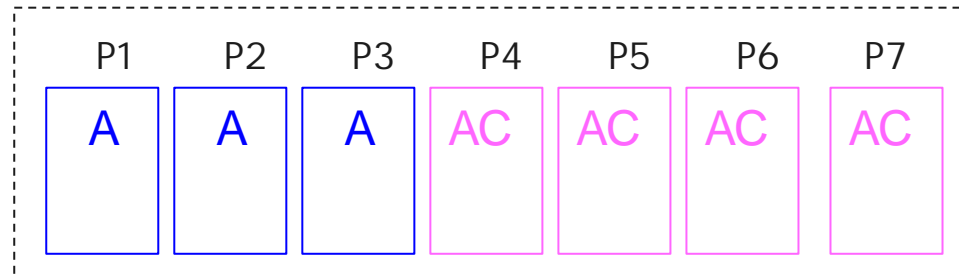
- > optimal for throughput if components are naturally parallel
- > hard to load balance
- > could result in a waste of resources on some platforms if models are sequential by construction

Many executables:

- > harder to manage for the OS (?)
- > no I/O or communicator conflicts
- > optimal for memory requirements
- > memory copy (MC) or memory sharing (MS) not possible
- > message passing possible: flexible, but message passing overhead

Discussion on possible coupled architecture

3. One executable with parallel execution of components on different processors



Parallel execution:

- > optimal for throughput if component naturally parallel with different distribution
- > hard to load balance
- > demanding in memory if no dynamic allocation in models

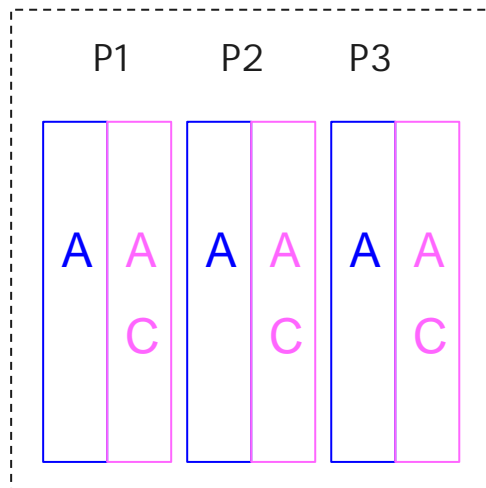
One executable on different processors:

- > easier to manage for the OS (?)
- > possible I/O and communicator conflicts
- > memory sharing or copy not possible
- > message passing possible

=> No real advantage and more disadvantages compared to 2.

Discussion on possible coupled architecture

4. One executable with parallel execution of components in threads



Parallel execution:

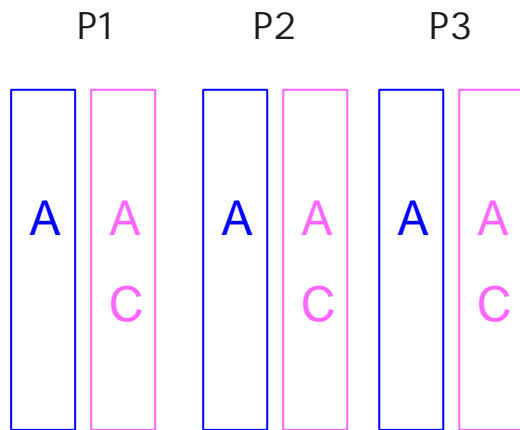
- > optimal for throughput
if component naturally parallel with same distribution

One executable:

- > easier to manage for the OS (?)
- > possible I/O and communicator conflicts
- > memory sharing possible (not flexible but efficient)
- > memory copy possible (flexible, less efficient than MS)
- > if A and AC perform external MPI communication (e.g. A ->O), MPI not possible between A and AC
- > if only A or AC needs external MPI, maybe it would work (cf H. Ritzdorf 25-10-2001)
- > On SGI IRIX, MPI and OpenMP are easy to combine => good alternative (cf R. Vogelsang 30-1-2002)

Discussion on possible coupled architecture

5. Many executables with parallel execution on same set of processors



Parallel execution:

-> optimal for throughput

if component naturally parallel with same distribution

Many executables:

-> harder to manage for the OS (?)

-> no I/O nor communicator conflicts

-> message passing possible

Same processors:

-> coupling through shared memory segments possible (portability?)

-> **difficult to guarantee use of same processors !**